Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-12 (2016) 640–644

Routing problems: constraints and optimality ⋆

Alexander G. Chentsov * **Pavel A. Chentsov** ** **Alexander A. Petunin** *** **Alexander N. Sesekin** ****

* *Ural Federal University, Russia. Institute of Mathematics and Mechanics UB RAS, Russia, (Tel: +7-343-375-34-57; e-mail: chentsov@imm.uran.ru).*

** *Ural Federal University, Russia. Institute of Mathematics and Mechanics UB RAS, Russia, (Tel: +7-343-362-81-62; e-mail: chentsov.p@uran.ru).*

*** *Ural Federal University, Russia, (Tel: +7-343-375-97-14; e-mail: aapetunin@gmail.com).*

**** *Ural Federal University, Russia, (Tel: +7-343-375-41-40; e-mail: sesekin@list.ru).*

Abstract: We consider the issues of routing under constraints and formulate a mathematical problem of visiting megalopolises. The order of visits is subject to precedence constraints. In addition, the cost functions depend on the set of pending tasks. The quality criterion is a variety of the additive criterion. The problem is established within the dynamic programming framework, however, a heuristic is proposed and implemented to solve practical problems of large dimensionality.

1. INTRODUCTION

There is a number of distinct applications where routing problems arise (e.g. transportation problems, atomic power generation, toolpath optimization for CNC cutting machines). In many problems, there is a need to order, or sequence, the operations. Often, the sequencing is complicated by various constraints. At times, it is more important to satisfy all the constraints than to achieve an optimum for the quality criterion—the typical case in real-life engineering.

We consider an optimization problem for the sake of making the search for admissible solutions more effective. To this end, we employ a pretty complicated variant of dynamic programming (DP) supplemented with special constructions which decrease the computational complexity though the use precedence constraints. It is natural to use DP in local way, which is implemented as *insertions* (or “incuts”) of moderate dimension. Basic constructions of this approach are reflected in Chentsov (2008a, 2014a,b). Of course, the above-mentioned mathematical problem has many applications.

It is important that the DP-based procedure we employ provides for optimal solutions of precedence constrained problems where the cost functions depend on the set of pending tasks. This is its main feature as far as the DP structure is concerned (the optional dependence of cost functions on the set of pending tasks being its most essential and innovative part). The main issue in using DP is the deficit of computer memory (DP is not a polynomial procedure), which makes its use infeasible for problems of large dimension. For those, we have to resort to heuristic algorithms.

2. PROBLEM STATEMENT AND GENERAL NOTATION

Let us fix an arbitrary nonempty set X, which will serve as the ground set. Let $x^0 \in X$ be the corresponding initial state, or the base (depot) of our process. Let N be a natural number for which $N \geq 2$. Fix nonempty finite sets $M_1, ..., M_N$ for which

$$M_1 \subset X, ..., M_N \subset X.$$

We regard these sets as megalopolises. Assume

$$x^0 \notin M_1, ..., x^0 \notin M_N;$$

moreover, the sets $M_1, ..., M_N$ are pairwise disjoint. The megalopolises must be visited in a certain sequence and, in each of them, a certain job has to be completed, which results into a kind of interior movement within each megalopolis. We consider the arrival point and the departure point. The pairs of these points are evaluated by the corresponding cost functions.

Next, each exterior permutation (from x^0 to megalopolises and between megalopolises) is evaluated. Finally, for terminal state, the corresponding evaluation is defined. We consider the case when all the costs are summed, the additive case. However, the optimization is complicated by the constraints, which restrict both the routes defined as permutations of the megalopolises' indices and the trajectories of specific motions. Specifically, we consider a process defined by the following scheme:

$$\begin{aligned} x^0 &\longrightarrow (x_{1,1} \in M_{\alpha(1)} \rightsquigarrow x_{1,2} \in M_{\alpha(1)}) \longrightarrow ... \\ ... &\longrightarrow (x_{N,1} \in M_{\alpha(N)} \rightsquigarrow x_{N,2} \in M_{\alpha(N)}), \end{aligned} \qquad (1)$$

⋆ This work was supported by Act 211 Government of the Russian Federation, contract N 02.A03.21.0006

10.1016/j.ifacol.2016.07.756

where α is a route, or indices' permutation. Constraints are imposed on α, $x_1 = (x_{1,1}, x_{1,2}), ..., x_{N-1} = (x_{N-1,1}, x_{N-1,2})$, and $x_N = (x_{N,1}, x_{N,2})$ (here, it is supposed that $N > 2$). In (1), the straight arrows denote the exterior movements and the sinuous arrows denote the movements connected with interior jobs. As mentioned above, all steps in (1) are evaluated through the corresponding cost functions. The sum of their costs defines the additive criterion.

Now, let us recall the constraints. We note only the most important. The choice of the route α in (1) can be constrained by precedence constraints. Namely, there may be a nonempty set of ordered sender-receiver pairs. The route α must order the megalopolises so as to make sure that each sender megalopolis appears before its corresponding receiver. These are the route constraints.

Now for trajectory constraints, which restrict $x_1, ..., x_N$ in (1). Previously, we remarked that, in many cases, this trajectory must be realized in the form

$$x_1 \in \mathcal{M}_{\alpha(1)}, ..., x_N \in \mathcal{M}_{\alpha(N)},$$

where $\mathcal{M}_j \subset M_j \times M_j$ for all $j \in \overline{1,N}$. Now, we note that $\mathcal{M}_j$ defines all possible variants of conducting the interior jobs connected with the megalopolis M_j, which are denoted by sinuous arrows in (1). In this connection, see the following example:

Example. Consider the problem of routing the tool in sheet cutting CNC machines. The tool arrives at the contour to cut it; it must arrive at a given penetration point near the equidistant of the contour, and it is the equidistant that is to be cut to create a desired contour—roughly speaking, the cut is made such that there is some space between the desired contour and the cut. After the contour is cut, the tool, while still active, must be driven to the corresponding shut-off point. It is important to observe precedence constraints: each interior (child) contour of a feature must be cut out before the exterior (parent) contour. There are also other constraints, for example, the sheet rigidity must be maintained at all times (a new cut may only be done in the rigidity zone), and the sheet must remain in one piece (i.e., not cut in two, etc.), which may be formalized as the dependence of the cost on the set of pending tasks. Thus there appears a kind of memory, or sequence dependence. Our principal aim in constructing the route and trajectory is to satisfy all those constraints.

The quality criterion is the total amount of time required to cut all the features. To provide for satisfaction of certain constraints, we may make the cost of "prohibited" actions effectively infinite; thus optimization would contribute to constraint satisfaction.

We propose a method for constructing a route subject to certain constraints that is related to DP. However, the mentioned constraints complicate the construction of the DP procedure.

3. EXTREMAL PROBLEM (GENERAL REMARKS)

In this section, we describe the mathematical setting of the routing problem with constraints. We follow the informative setting of the previous section. In addition, we assume that $\mathcal{N} \triangleq \{1; 2; ...\}$, $\mathcal{N}_0 \triangleq \{0\} \bigcup \mathcal{N} = \{0; 1; 2; ...\}$, and, for $p \in \mathcal{N}_0$ and $q \in \mathcal{N}_0$, $\overline{p,q} \triangleq \{i \in \mathcal{N}_0 \mid (p \leq i) \& (i \leq q)\}$ (of course, $\overline{p,q} = \emptyset$ under $q < p$). For a nonempty set H, denote by $\mathcal{R}_+[H]$ set of all nonnegative real-valued functions defined on H. Let $\mathbf{M}_j \triangleq \{\mathrm{pr}_2(z) : z \in \mathcal{M}_j\}$ for $j \in \overline{1,N}$ ($\mathrm{pr}_1(h)$ and $\mathrm{pr}_2(h)$ denote the first and second elements of an arbitrary ordered pair h). Let

$$\mathcal{X} \triangleq \{x^0\} \bigcup (\bigcup_{i=1}^{N} M_i), \mathbf{X} \triangleq \{x^0\} \bigcup (\bigcup_{i=1}^{N} \mathbf{M}_j),$$

$\mathbf{X} \subset \mathcal{X} \subset X$. In addition, $\mathcal{X}$ and $\mathbf{X}$ are nonempty finite sets.

Consider the possible variants of precedence constraints. Fix the set $\mathbf{K}$ with the property $\mathbf{K} \subset \overline{1,N} \times \overline{1,N}$. For every nonempty set $\mathbf{K}_0$, $\mathbf{K}_0 \subset \mathbf{K}$, assume

$$\exists z_0 \in \mathbf{K}_0 : \mathrm{pr}_1(z_0) \neq \mathrm{pr}_2(z) \ \ \forall z \in \mathbf{K}_0; \tag{2}$$

see (Chentsov , 2008a, Condition 2.2.1) (condition (2) is fulfilled in many practically interesting cases; see (Chentsov , 2008a, ch.2)). Then,

$$\begin{aligned} \mathbf{A} \triangleq \{\alpha \in \mathbf{P} \mid \forall z \in \mathbf{K} \, \forall t_1 \in \overline{1,N} \, \forall t_2 \in \overline{1,N} \\ ((\alpha(t_1) = \mathrm{pr}_1(z)) \& (\alpha(t_2) = \mathrm{pr}_2(z))) \Rightarrow (t_1 < t_2)\} = \\ = \{\alpha \in \mathbf{P} \mid \alpha^{-1}(\mathrm{pr}_1(z)) < \alpha^{-1}(\mathrm{pr}_2(z)) \ \ \forall z \in \mathbf{K}\} \neq \emptyset, \end{aligned} \tag{3}$$

where $\mathbf{P}$ is the set of all permutations of $\overline{1,N}$ and, for every $\beta \in \mathbf{P}$, the symbol β^{-1} denotes the inverse of the permutation β. Thus, an admissible (in view of precedence constraints) route exists.

By $\tilde{\mathbf{Z}}$ we denote the set of all tuples $(z_i)_{i \in \overline{0,N}} : \overline{0,N} \longrightarrow \mathcal{X} \times \mathcal{X}$ and

$$\begin{aligned} \mathbf{Z}_\alpha \triangleq \{(z_i)_{i \in \overline{0,N}} \in \tilde{\mathbf{Z}} \mid (z_0 = (x^0, x^0)) \& (z_t \in \mathcal{M}_{\alpha(t)} \\ \forall t \in \overline{1,N})\} \neq \emptyset \ \ \forall \alpha \in \mathbf{A}. \end{aligned} \tag{4}$$

In (4), the trajectories coordinated with a route are introduced. Then,

$$\mathbf{D} \triangleq \{(\alpha, (z_i)_{i \in \overline{0,N}}) \in \mathbf{A} \times \tilde{\mathbf{Z}} \mid (z_i)_{i \in \overline{0,N}} \in \mathbf{Z}_\alpha\} \neq \emptyset$$

is the set of all admissible solutions.

Cost functions. In the following, we consider an "additive" routing problem similar to Chentsov (2014a,b) (a variant of bottleneck problem was considered in Chentsov (2008b, 2015) and in other articles). Now, we introduce the costs of movements, interior jobs, and the terminal state. Denote by $\mathbf{N}$ the family of all nonempty subsets of $\overline{1,N}$. Fix $\mathbf{c} \in \mathcal{R}_+[\mathcal{X} \times \mathcal{X} \times \mathbf{N}]$, $c_1 \in \mathcal{R}_+[\mathcal{X} \times \mathcal{X} \times \mathbf{N}]$,..., $c_N \in \mathcal{R}_+[\mathcal{X} \times \mathcal{X} \times \mathbf{N}]$ and $f \in \mathcal{R}_+[\mathcal{X}]$.

Mathematical setting of problem. If $\alpha \in \mathbf{A}$ and $(z_t)_{i \in \overline{0,N}} \in \mathbf{Z}_\alpha$, then

$$\begin{aligned} \mathbf{C}_\alpha[(z_t)_{t \in \overline{0,N}}] \triangleq \sum_{t=1}^{N} [\mathbf{c}(\mathrm{pr}_2(z_{t-1}), \mathrm{pr}_1(z_t), \\ \{\alpha(j) : j \in \overline{t,N}\}) + c_{\alpha(t)}(z_t, \{\alpha_j : j \in \overline{t,N}\})] + \\ + f(\mathrm{pr}_2(z_N)) \end{aligned} \tag{5}$$

(recall that, for three nonempty sets A, B and C, $A \times B \times C = (A \times B) \times C$; in particular, $\mathcal{X} \times \mathcal{X} \times \mathbf{N} = (\mathcal{X} \times \mathcal{X}) \times \mathbf{N}$). Then, our basic problem is formulated as

$$\mathbf{C}_\alpha[(z_t)_{t \in \overline{0,N}}] \longrightarrow \min, (\alpha, (z_t)_{t \in \overline{0,N}}) \in \mathbf{D}. \tag{6}$$

4. ALGORITHM ON FUNCTIONAL LEVEL

To solve (6), in Chentsov (2008a, 2014a,b) (and in other articles), an economic variant of DP was constructed. On this basis, algorithms were developed and implemented on PC. Of course, these algorithms can be applied for problems of moderate dimensions. But it is also possible to employ them in construction of optimized insertions and iterated procedures making use of such insertions; see Chentsov (2014c); Petunin (2014). In connection with the problem concerning sheet cutting on the CNC machines, we note Petunin (2009); Frolovskij (2005).

Some questions on tool path route optmization for CNC cutting machines were considered in Hoeft (1997); Dewil (2011, 2014); Xie (2009); Yang (2010). However, unlike the proposed mathematical model, this works do not take into account the dependence of cost functions on the list of completed jobs, which is important for technological restrictions compliance (Petunin (2015)).

General questions of solution of traveling salesman problem (TSP) are considered in Melamed (1989a,b,c); Gutin (2002), and many other publications. In connection with the application of DP for solution of TSP, we note Bellman (1958); Held (1962).

Let us briefly sketch the procedure. To this end, we introduce the operator $\mathbf{I}$ of (Chentsov , 2008a, Ch.2), $\mathbf{I}:\mathbf{N}\longrightarrow\mathbf{N}$. Namely, for $K\in\mathbf{N}$, let

$$\mathbf{I}(K)\stackrel{\triangle}{=}K\setminus\{\mathrm{pr}_2(z):z\in\Xi[K]\},$$

where $\Xi[K]\stackrel{\triangle}{=}\{z\in\mathbf{K}\mid(\mathrm{pr}_1(z)\in K)\&(\mathrm{pr}_2(z)\in K)\}$.

Admissible task lists. Set $\mathcal{G}\stackrel{\triangle}{=}\{K\in\mathbf{N}\mid\forall z\in\mathbf{K}\ (\mathrm{pr}_1(z)\in K)\Rightarrow(\mathrm{pr}_2(z)\in K)\}$; moreover, for $s\in\overline{1,N}$, let $\mathcal{G}_s\stackrel{\triangle}{=}\{K\in\mathcal{G}\mid s=|K|\}$, where, for a nonempty finite set $\mathbf{F}$, its cardinality is denoted by $|\mathbf{F}|$. Of course, $\mathcal{G}_N=\{\overline{1,N}\}$ (the singleton containing the set $\overline{1,N}$). Let $\mathbf{K}_1\stackrel{\triangle}{=}\{\mathrm{pr}_1(z):z\in\mathbf{K}\}$; then, $\mathcal{G}_1=\{\{t\}:t\in\overline{1,N}\setminus\mathbf{K}_1\}$. Finally,

$$\mathcal{G}_{s-1}=\{K\setminus\{t\}:K\in\mathcal{G}_s,t\in\mathbf{I}(K)\}\ \ \forall s\in\overline{2,N}.$$

We obtain a natural step-by-step procedure: $\mathcal{G}_N\longrightarrow\mathcal{G}_{N-1}\longrightarrow...\longrightarrow\mathcal{G}_1$ (if $N>2$, obviously).

Layers of position space. An ordered pair (x,K), where $x\in\mathbf{X}$ and $K\in\mathbf{N}\bigcup\{\emptyset\}$, is considered a position. We construct sets $D_0,D_1,...,D_N$ in the state space. Suppose that $\tilde{\mathcal{M}}$ is the union of all sets $\mathbf{M}_j, j\in\overline{1,N}\setminus\mathbf{K}_1$. Then, $D_0\stackrel{\triangle}{=}\{(x,\emptyset):x\in\tilde{\mathcal{M}}\}$ and $D_N\stackrel{\triangle}{=}\{(x^0,\overline{1,N})\}$ (a singleton containing the ordered pair $(x^0,\overline{1,N})$). Consider the construction procedure for the set $D_{\mathbf{s}}$, where $\mathbf{s}\in\overline{1,N-1}$. For $K\in\mathcal{G}_{\mathbf{s}}$, let

$$\mathcal{J}_{\mathbf{s}}(K)\stackrel{\triangle}{=}\{j\in\overline{1,N}\setminus K\mid\{j\}\bigcup K\in\mathcal{G}_{\mathbf{s}+1}\},\ \hat{\mathcal{M}}_{\mathbf{s}}[K]\stackrel{\triangle}{=}\stackrel{\triangle}{=}\bigcup_{j\in\mathcal{J}_s(K)}\mathbf{M}_j,$$

and let $\mathcal{D}_{\mathbf{s}}[K]\stackrel{\triangle}{=}\{(x,K):x\in\hat{\mathcal{M}}_{\mathbf{s}}[K]\}$. Finally, let

$$D_{\mathbf{s}}\stackrel{\triangle}{=}\bigcup_{K\in\mathcal{G}_{\mathbf{s}}}\mathcal{D}_{\mathbf{s}}[K].$$

Thus we obtain the required sets $D_j, j\in\overline{0,N}$. For a more detailed construction, refer to (Chentsov , 2008a, §4.9). If $s\in\overline{1,N}$, $(x,K)\in D_s$, $j\in\mathbf{I}(K)$, and $z\in\mathcal{M}_j$, then

$$(\mathrm{pr}_2(z),K\setminus\{j\})\in D_{s-1}.$$

We note that $D_0\neq\emptyset$, $D_1\neq\emptyset$,..., $D_N\neq\emptyset$.

The Layers of the Bellman function. We are going to describe the following procedure of construction of the real-valued functions:

$$v_0\longrightarrow v_1\longrightarrow...\longrightarrow v_N.$$

Let $v_0\in\mathcal{R}_+[D_0]$ and let

$$v_0(x,\emptyset)\stackrel{\triangle}{=}f(x)\ \ \forall x\in\tilde{\mathcal{M}}.$$

Let $s\in\overline{1,N}$ and suppose v_{s-1} is already constructed. Then, $v_s\in\mathcal{R}_+[D_s]$ is defined by the rule

$$\begin{aligned}v_s(x,K)\stackrel{\triangle}{=}\min_{j\in\mathbf{I}(K)}\min_{z\in\mathcal{M}_j}[\mathbf{c}(x,\mathrm{pr}_1(z),K)+\\+c_j(z,K)+v_{s-1}(\mathrm{pr}_2(z),K\setminus\{j\})]\ \ \forall(x,K)\in D_s.\end{aligned}\tag{7}$$

Thus we obtain the recurrent procedure $v_0\longrightarrow v_1\longrightarrow...\longrightarrow v_N$. In addition, $V\stackrel{\triangle}{=}v_N(x^0,\overline{1,N})\in[0,\infty[$ is realized. Moreover, V is the global extremum for our routing problem (6); namely,

$$V=\min_{(\alpha,\mathbf{z})\in\mathbf{D}}\mathbf{C}_\alpha[\mathbf{z}]=\min_{\alpha\in\mathbf{A}}\min_{(z_t)_{t\in\overline{0,N}}\in\mathbf{Z}_\alpha}\mathbf{C}_\alpha[(z_t)_{t\in\overline{0,N}}].\tag{8}$$

Optimal solution. From (7) and (8), we obtain

$$\begin{aligned}V=\min_{j\in\mathbf{I}(\overline{1,N})}\min_{z\in\mathcal{M}_j}[\mathbf{c}(x^0,\mathrm{pr}_1(z),\overline{1,N})+\\+c_j(z,\overline{1,N})+v_{N-1}(\mathrm{pr}_2(N),\overline{1,N}\setminus\{j\})].\end{aligned}\tag{9}$$

Let $\mathbf{z}_0\stackrel{\triangle}{=}(x^0,x^0)$. Using (9), we choose $\eta_1\in\mathbf{I}(\overline{1,N})$ and $\mathbf{z}^{(1)}\in\mathcal{M}_{\eta_1}$ for which

$$\begin{aligned}V=\mathbf{c}(x^0,\mathrm{pr}_1(\mathbf{z}^{(1)}),\overline{1,N})+c_{\eta_1}(\mathbf{z}^{(1)},\overline{1,N})+\\+v_{N-1}(\mathrm{pr}_2(\mathbf{z}^{(1)}),\overline{1,N}\setminus\{\eta_1\}).\end{aligned}\tag{10}$$

Then, $(\mathrm{pr}_2(\mathbf{z}^{(1)}),\overline{1,N}\setminus\{\eta_1\})\in D_{N-1}$ and, by (7),

$$\begin{aligned}v_{N-1}(\mathrm{pr}_2(\mathbf{z}^{(1)}),\overline{1,N}\setminus\{\eta_1\})=\\=\min_{j\in\mathbf{I}(\overline{1,N}\setminus\{\eta_1\})}\min_{z\in\mathcal{M}_j}[\mathbf{c}(\mathrm{pr}_2(\mathbf{z}^{(1)}),\mathrm{pr}_1(z),\overline{1,N}\setminus\{\eta_1\})+\\+c_j(z,\overline{1,N}\setminus\{\eta_1\})+v_{N-2}(\mathrm{pr}_2(z),\overline{1,N}\setminus\{\eta_1;j\})].\end{aligned}$$

We choose $\eta_2\in\mathbf{I}(\overline{1,N}\setminus\{\eta_1\})$ and $\mathbf{z}^{(2)}\in\mathcal{M}_{\eta_2}$ such that

$$\begin{aligned}v_{N-1}(\mathrm{pr}_2(\mathbf{z}^{(1)}),\overline{1,N}\setminus\{\eta_1\})=\\=\mathbf{c}(\mathrm{pr}_2(\mathbf{z}^{(1)}),\mathrm{pr}_1(\mathbf{z}^{(2)}),\overline{1,N}\setminus\{\eta_1\})+\\+c_{\eta_2}(\mathbf{z}^{(2)},\overline{1,N}\setminus\{\eta_1\})+\\+v_{N-2}(\mathrm{pr}_2(\mathbf{z}^{(2)}),\overline{1,N}\setminus\{\eta_1,\eta_2\})\in D_{N-2}.\end{aligned}\tag{11}$$

From (10) and (11), we obtain

$$\begin{aligned}V=\mathbf{c}(x^0,\mathrm{pr}_1(\mathbf{z}^{(1)}),\overline{1,N})+\\+\mathbf{c}(\mathrm{pr}_2(\mathbf{z}^{(1)}),\mathrm{pr}_1(\mathbf{z}^{(2)}),\overline{1,N}\setminus\{\eta_1\})+c_{\eta_1}(\mathbf{z}^{(1)},\overline{1,N})+\\+c_{\eta_2}(\mathbf{z}^{(2)},\overline{1,N}\setminus\{\eta_1\})+v_{N-2}(\mathrm{pr}_2(\mathbf{z}^{(2)}),\overline{1,N}\setminus\{\eta_1;\eta_2\}).\end{aligned}$$

This procedure should be continued until the whole list of tasks is exhausted. As a result, we obtain the route

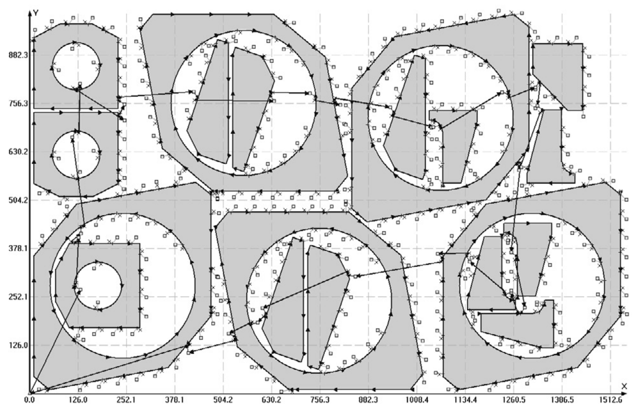

Fig. 1. Sample 1. Dynamic programming solution.

$\eta \stackrel{\triangle}{=} (\eta_j)_{j\in\overline{1,N}} \in \mathbf{A}$ and trajectory $(\mathbf{z}^{(j)})_{j\in\overline{0,N}} \in \mathcal{Z}_\eta$ for which

$$\mathcal{C}_\eta[(\mathbf{z}^{(j)})_{j\in\overline{0,N}}] = V.$$

Thus, $(\eta, (\mathbf{z}^{(j)})_{j\in\overline{0,N}}) \in \mathbf{D}$ is the required optimal solution (a more detailed presentation of this scheme is contained in Chentsov (2014a)).

5. COMPUTATIONAL EXPERIMENT

Calculations were made on the computer with the Intel i7-2630QM processor, 8GB memory, and the Windows 7 (64-bit) operating system. For small parts, these numbers could be reduced.

There is a much data on the coordinates of points and the resulting route; we omit these due to space constraints. All external cost functions are distances between points. Internal cost functions depend on the sets of pending tasks.

Example 1. Number of megalopolises $N = 27$. Dynamic programming method. The calculation took 1 hour 22 min. 41 sec. The value obtained was 89.07.

We again note that DP may reasonably be applied to the problems with moderate values of N ($N \approx 31$). For problems of larger dimensions, it is possible to use DP to construction local insertions; this possibility was studied in Chentsov A.A., Chentsov A.G. 2 (2014) and Petunin A.A., Chentsov A.G. and Chentsov P.A. (2014). On this basis, we can construct iterative procedures.

6. HEURISTIC ALGORITHM

Now, we consider a variant of heuristic algorithm for solving "big" constrained routing problems. To account for some natural constraints, we use cost functions with dependence on the set of pending tasks. However, it is quite hard to implement such cost functions from the computational perspective (the complexity is especially pronounced for cost functions with dependence on the set of pending tasks; however, such cost functions are important for complying with the above-mentioned constraints). The proposed algorithm does not require a construction of "full" cost functions. The corresponding calculations of the required values of cost functions are done as a specific trajectory of a process develops. Using the fragments of these functions, we implement the local greedy choice of the next task.

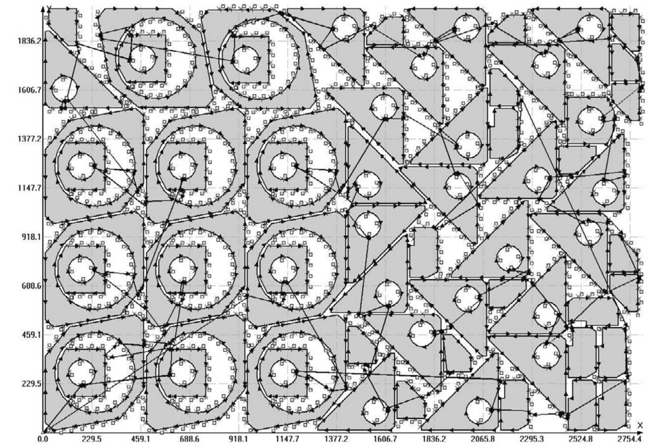

Fig. 2. Sample 2. Heuristic solution.

This idea is conceptually similar to the feedback control: calculations are realized for a fully developed position. This is an original approach to solution of large-scale problems, for which DP is computationally infeasible.

Moreover, in the given algorithm, system improving corrections are provided; in addition, the corresponding exclusions on permutations are used.

Heuristic algorithm consists of several steps. It creates a route by adding the contours step by step. Every next contour to be added to the route is obtained as the one that provides the minimum cost (internal cost functions depend on the preceding part of the route) and satisfies precedence constraints.

Example 2. Number of megalopolises $N = 112$. Heuristic method. The calculation took 1 min. 2 sec. The value obtained was 383.98.

For Example 1, the heuristic method obtained the value 89.35. It is close to the DP result. It is the reason why the heuristic algorithm may be used for problems of large dimension.

7. CONCLUSION

The considered class of problems has essential singularities that were not considered previously. The principal singularity of this type is connected with the possible dependence of cost functions on the set of pending tasks, which might be necessary to satisfy all the necessary constraints arising in real-life problems and, in particular, in sheet cutting problems for CNC machines. Such means of accounting for the mentioned constraints is a new contribution to the field. It is important that, in this case, the cost functions depend on the set of pending tasks. The authors know of only two works concerned with solving a routing problem where the costs depend on the set of pending tasks, Alkaya (2015); Leon (1996). However, in these papers, there was only considered a very particular case of the TSP with a complicated cost matrix, for which heuristic algorithms were proposed; optimal solution was not attempted. It is not possible to apply those algorithms to the problem our paper is concerned with (we deal with a greater number of constraints and, in particular, the precedence constraints). A particular obstacle is the real-time computation of complicated cost functions. The problem considered in the paper is much more complex, and much more specialized algorithms are necessary to solve it. Such an algorithm was presented in Section 6. We note that a DP-based procedure can be applied to testing the heuristics for this problem on sample instances of limited dimension. DP and heuristic procedures ought to be considered together as we did in this paper.

8. ACKNOWLEDGMENTS

This work was supported by Act 211 Government of the Russian Federation, contract N 02.A03.21.0006.

REFERENCES

Ali Fuat Alkaya, Ekrem Duman Combining and solving sequence dependent traveling salesman and quadratic assignment problems in PCB assembly Discrete Applied Mathematics, 10 September 2015, Vol. 192, p. 2-16.

Bellman R. On a Routing Problem Quart. Appl. Math, 16, 1958, P.87-90.

Chentsov A.G. Ekstremal'nye zadachi marshrutizatsii i raspredeleniya zadaniy: voprosy teorii. M.-Izhevsk: NITs "Regulyarnaya i khaoticheskaya dinamika", Izhevskiy institut komp'yuternykh issledovaniy, 2008, 240 s.

Chentsov A.A., Chentsov A.G. Extremal Bottleneck Routing Problem with Constraints in the Form of Precedence Conditions. Proceedings of the Steklov Institute of Mathematics (Supplementary issues), 2008, vol. 263, no.2 p.23-36.

Chentsov A.G. Problem of successive megalopolis traversal with the precedence conditions. Automation and Remote Control, April 2014, Volume 75, Issue 4, p. 728-744.

Chentsov A.A., Chentsov A.G., Chentsov P.A. Elements of dynamic programming in extremal routing problems. Automation and Remote Control, Volume 75, Issue 3, March 2014, p. 537-550.

Chentsov A.A., Chentsov A.G. Zadacha posledovatel'nogo obhoda megapolisov. Vestn. Tambov. un-ta. Ser. Estestv. i tehn. nauki, 2014, T. 19, vyp. 2, S.454-475.

Chentsov A.G., Salii Ya.V. A Model of "Nonadditive" Routing Problem Where the Costs Depend on the Set of Pending Tasks. Bulletin SUSU MMCS, Mathematical Modelling Programming & Computer Software, 2015, vol.8, no.1. p. 24-44.

Dewil R., Vansteenwegen P., Cattrysse D. Cutting Path Optimization using Tabu Search. Key Engineering Materials, 473, 2011. p. 739-748.

Dewil R., Vansteenwegen P., Cattrysse D. Construction heuristics for generating tool paths for laser cutters. International Journal of Production Research, Mar. 2014, p. 1-20.

Frolovskij V.D. Avtomatizacija proektirovanija upravljajushhih programm teplovoj rezki metalla na oborudovanii s ChPU Informacionnye tehnologii v proektirovanii i proizvodstve. N4. M. 2005. S. 63-66.

Gutin G., Punnen A. The Traveling Salesman Problem and Its Variations. Berlin: Springer, 2002, 850 p.

Held M., Karp R.M. A Dynamic Programming Approach to Sequencing Problems Journal of the Society for Industrial and Applied Mathematics, 1962, N10 (1), P. 196-210.

Hoeft J., Palekar U.S. Heuristics for the plate-cutting traveling salesman problem. IIE Transactions, 29, 1997, p. 719-731.

V.Jorge Leon, Brett A.Peters Replanning and Analysis of Partial Setup Strategies in Printed Circuit Board Assembly Systems The International Journal of Flexible Manufacturing Systems, 1996, N8, p. 389-412.

Melamed I.I., Sergeev S.I., Sigal I.Kh. The traveling salesman problem. Issues in theory Automation and Remote Control, 1989, 50:9, P. 1147-1173.

Melamed I.I., Sergeev S.I., Sigal I.Kh. The traveling salesman's problem. Exact methods. Automation and Remote Control, 1989, 50:10, P. 1303-1324.

Melamed I.I., Sergeev S.I., Sigal I.Kh. The traveling salesman problem. Approximate algorithms. Automation and Remote Control, 1989, 50:11, P. 1459-1479.

Petunin A.A. O nekotoryh strategijah formirovanija marshruta instrumenta pri razrabotke upravljajushhih programm dlja mashin termicheskoj rezki metalla. Vestnik UGATU. 2009, T.13, N.2, p.280-286.

Petunin A.A., Chentsov A.G., Chentsov P.A. Local dynamic programming incuts in routing problems with restrictions. Vestn. Udmurtsk. Univ. Mat. Mekh. Komp. Nauki, 2014, N2, p. 56-75.

Petunin A.A. Modelling of Tool Path for the CNC Sheet Cutting Machines. AIP Conference Proceedings. 41st International Conference Applications of Mathematics in Engineering and Economics (AMEE'15), 2015, V.1690, p. 060002(1)-060002(7).

Xie S.Q., et al. Optimal process planning for compound laser cutting and punch using Genetic Algorithms. International Journal of Mechatronics and Manufacturing Systems, 2009, 2 (1/2), p.20-38.

Yang W.B., et al. An Effective Algorithm for Tool-Path Airtime Optimization during Leather Cutting. Advanced Materials Research, 2010, 102-104, 373-377.